

Documentação do Sistema de Controle de Temperatura do Aviário

RAMON MARQUES

1. Descrição Geral

Este código implementa um sistema de controle de temperatura para um aviário usando lógica fuzzy. O objetivo do sistema é controlar automaticamente a temperatura interna do aviário levando em consideração várias variáveis relevantes, como temperatura, umidade e idade das aves.

1.2. Variáveis

O sistema leva em consideração as seguintes variáveis:

- Temperatura: temperatura atual do aviário medida em graus Celsius.
- Umidade: umidade atual do aviário medida em porcentagem.
- Aquecimento necessário: indica se o aquecimento é necessário (sim ou não).
- Idade: idade das aves em dias.

1.3. Saídas

O sistema calcula as seguintes saídas:

- Ventilação: nível de ventilação necessário medido em porcentagem.
- Aquecimento: nível de aquecimento necessário medido em porcentagem.
- Tempo de funcionamento: tempo de funcionamento do sistema medido em minutos.

1.4. Funções de Pertinência

O sistema usa funções de pertinência triangular para definir como cada variável afeta as saídas. As funções de pertinência são definidas da seguinte maneira:

- Temperatura: baixa (0 a 22°C), ideal (18 a 30°C) e alta (28 a 44°C).
- Umidade: baixa (0 a 50%), ideal (30 a 70%) e alta (50 a 100%).
- Idade: jovem (0 a 15 dias) e adulto (15 a 21 dias).
- Aquecimento necessário: não e sim.
- Ventilação: desligado (0 a 50%), baixo (0 a 100%) e alto (50 a 100%).
- Aquecimento: baixo (0 a 50%), médio (0 a 100%) e alto (50 a 100%).
- Tempo de funcionamento: curto (0 a 30 minutos), médio (0 a 60 minutos) e longo (30 a 60 minutos).

1.5. Regras

O sistema usa as seguintes regras para calcular as saídas com base nas entradas:

1ª	Se a temperatura for	Baixa	E a umidade for	Baixa	Então a ventilação deve ser desligada.
2ª	Se a temperatura for	Baixa	E a umidade for	Ideal	Então a ventilação deve ser desligada.
3ª	Se a temperatura for	Baixa	E a umidade for	Alta	Então a ventilação deve ser baixa.
4ª	Se a temperatura for	Ideal	E a umidade for	Baixa	Então a ventilação deve ser desligada.
5ª	Se a temperatura for	Ideal	E a umidade for	Ideal	Então a ventilação deve ser desligada.
6ª	Se a temperatura for	Ideal	E a umidade for	Alta	Então a ventilação deve ser baixa.
7ª	Se a temperatura for	Alta	E a umidade for	Baixa	Então a ventilação deve ser alta.
8ª	Se a temperatura for	Alta	E a umidade for	Ideal	Então a ventilação deve ser alta.
9ª	Se a temperatura for	Alta	E a umidade for	Alta	Então a ventilação deve ser alta.
10ª	Se a temperatura for	Baixa	E a umidade for	Baixa	Então o tempo de funcionamento deve ser curto.
11ª	Se a temperatura for	Baixa	E a umidade for	Ideal	Então o tempo de funcionamento deve ser curto.
12ª	Se a temperatura for	Baixa	E a umidade for	Alta	Então o tempo de funcionamento deve ser médio.
13ª	Se a temperatura for	Ideal	E a umidade for	Baixa	Então o tempo de funcionamento deve ser curto.
14ª	Se a temperatura for	Ideal	E a umidade for	Ideal	Então o tempo de funcionamento deve ser curto.
15ª	Se a temperatura for	Ideal	E a umidade for	Alta	Então o tempo de funcionamento deve ser médio.
16ª	Se a temperatura for	Alta	E a umidade for	Baixa	Então o tempo de funcionamento deve ser longo.
17ª	Se a temperatura for	Alta	E a umidade for	Ideal	Então o tempo de funcionamento deve ser longo.
18ª	Se a temperatura for	Alta	E a umidade for	Alta	Então o tempo de funcionamento deve ser longo.

1.6. Uso

Para usar o sistema de controle de temperatura do aviário, basta fornecer as entradas (temperatura, umidade, idade e aquecimento necessário) e calcular as saídas (ventilação, aquecimento e tempo de funcionamento) usando o sistema de controle fuzzy implementado no código.

RAMON MARQUES

2. Especificação de Requisitos do Sistema de Controle de Temperatura do Aviário

2.1. Objetivo

O objetivo deste sistema é controlar automaticamente a temperatura interna de um aviário usando lógica fuzzy para levar em consideração várias variáveis relevantes, como temperatura, umidade e idade das aves.

2.2. Entradas

O sistema deve aceitar as seguintes entradas:

- Temperatura: temperatura atual do aviário medida em graus Celsius.
- Umidade: umidade atual do aviário medida em porcentagem.
- Idade: idade das aves em dias.
- Aquecimento necessário: indica se o aquecimento é necessário (sim ou não).

2.3. Saídas

O sistema deve calcular as seguintes saídas:

- Ventilação: nível de ventilação necessário medido em porcentagem.
- Aquecimento: nível de aquecimento necessário medido em porcentagem.
- Tempo de funcionamento: tempo de funcionamento do sistema medido em minutos.

2.4. Restrições e Limitações

- O sistema deve ser implementado em Python usando a biblioteca skfuzzy para lógica fuzzy.
- O sistema deve ser capaz de lidar com entradas incompletas ou incertas usando lógica fuzzy.
- O sistema deve ser capaz de fornecer saídas precisas e consistentes com base nas entradas fornecidas.

3. Diagrama de Arquitetura de Sistema

Infelizmente, como sou apenas um assistente de texto, não posso criar diagramas visuais. No entanto, posso descrever como um diagrama de arquitetura para um sistema de controle de temperatura em um aviário usando lógica fuzzy em Python pode ser estruturado.

Um diagrama de arquitetura para esse tipo de sistema pode incluir os seguintes componentes:

- Interface do usuário: Este componente representa a interface que os usuários usam para interagir com o sistema e fornecer entradas (temperatura, umidade, idade e aquecimento necessário).
- Módulo de lógica fuzzy: Este componente representa o módulo que implementa a lógica fuzzy para calcular as saídas (ventilação, aquecimento e tempo de funcionamento) com base nas entradas fornecidas pelo usuário.
- Módulo de controle: Este componente representa o módulo que controla os dispositivos de ventilação e aquecimento com base nas saídas calculadas pelo módulo de lógica fuzzy.

Esses componentes podem ser organizados em um diagrama visual que mostra como eles interagem entre si. Por exemplo, a interface do usuário pode enviar entradas para o módulo de lógica fuzzy, que calcula as saídas e as envia para o módulo de controle, que controla os dispositivos de ventilação e aquecimento.

Espero que esta descrição ajude você a entender como um diagrama de arquitetura para um sistema de controle de temperatura em um aviário usando lógica fuzzy em Python pode ser estruturado. Se você precisar criar um diagrama visual, pode usar uma ferramenta de desenho ou diagramação para representar os componentes e suas interações.

4. Plano de Teste do Sistema de Controle de Temperatura do Aviário

4.1. Objetivo

O objetivo deste plano de teste é verificar se o sistema de controle de temperatura do aviário está funcionando corretamente e atendendo aos requisitos especificados.

4.2. Requisitos

O sistema deve atender aos seguintes requisitos:

- Aceitar entradas para temperatura, umidade, idade e aquecimento necessário.
- Calcular saídas para ventilação, aquecimento e tempo de funcionamento com base nas entradas fornecidas.
- Produzir saídas precisas e consistentes com base nas entradas fornecidas.

4.3. Testes

Para verificar se o sistema está atendendo aos requisitos especificados, os seguintes testes serão realizados:

1. **Teste de Entrada:** Verificar se o sistema aceita entradas para temperatura, umidade, idade e aquecimento necessário.
2. **Teste de Saída:** Verificar se o sistema calcula saídas para ventilação, aquecimento e tempo de funcionamento com base nas entradas fornecidas.
3. **Teste de Precisão:** Verificar se o sistema produz saídas precisas com base nas entradas fornecidas.
4. **Teste de Consistência:** Verificar se o sistema produz saídas consistentes quando as mesmas entradas são fornecidas várias vezes.

4.4. Procedimentos

Para realizar os testes, os seguintes procedimentos serão seguidos:

1. Preparar um conjunto de dados de teste com diferentes combinações de entradas para temperatura, umidade, idade e aquecimento necessário.
2. Executar o sistema com cada conjunto de dados de teste e registrar as saídas produzidas.
3. Comparar as saídas produzidas com as saídas esperadas para verificar se o sistema está produzindo resultados precisos e consistentes.
4. Analisar os resultados dos testes e identificar quaisquer problemas ou inconsistências.

RAMON MARQUES